CMPs: *Now* and into the *Future*

Chuck Moore
AMD Senior Fellow
10/12/05

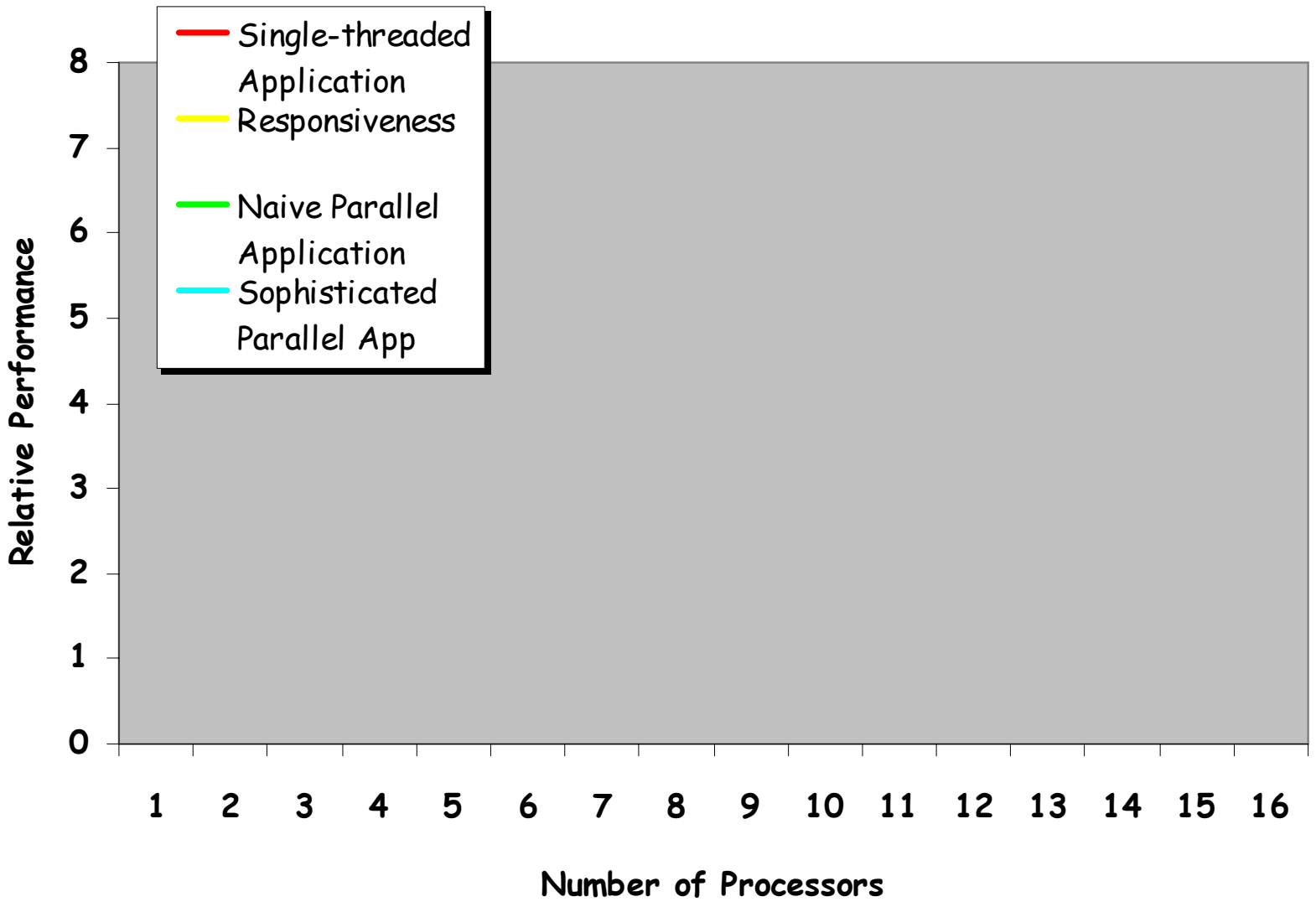
My Five Minutes at a glance

- ***Now...***
  - CMP Performance -- *reality check*
  - The transition to parallel applications
  - HW -- *key Issues and considerations*

- ***...and into the Future***
  - Generations 1-3
  - Generation 4 and beyond

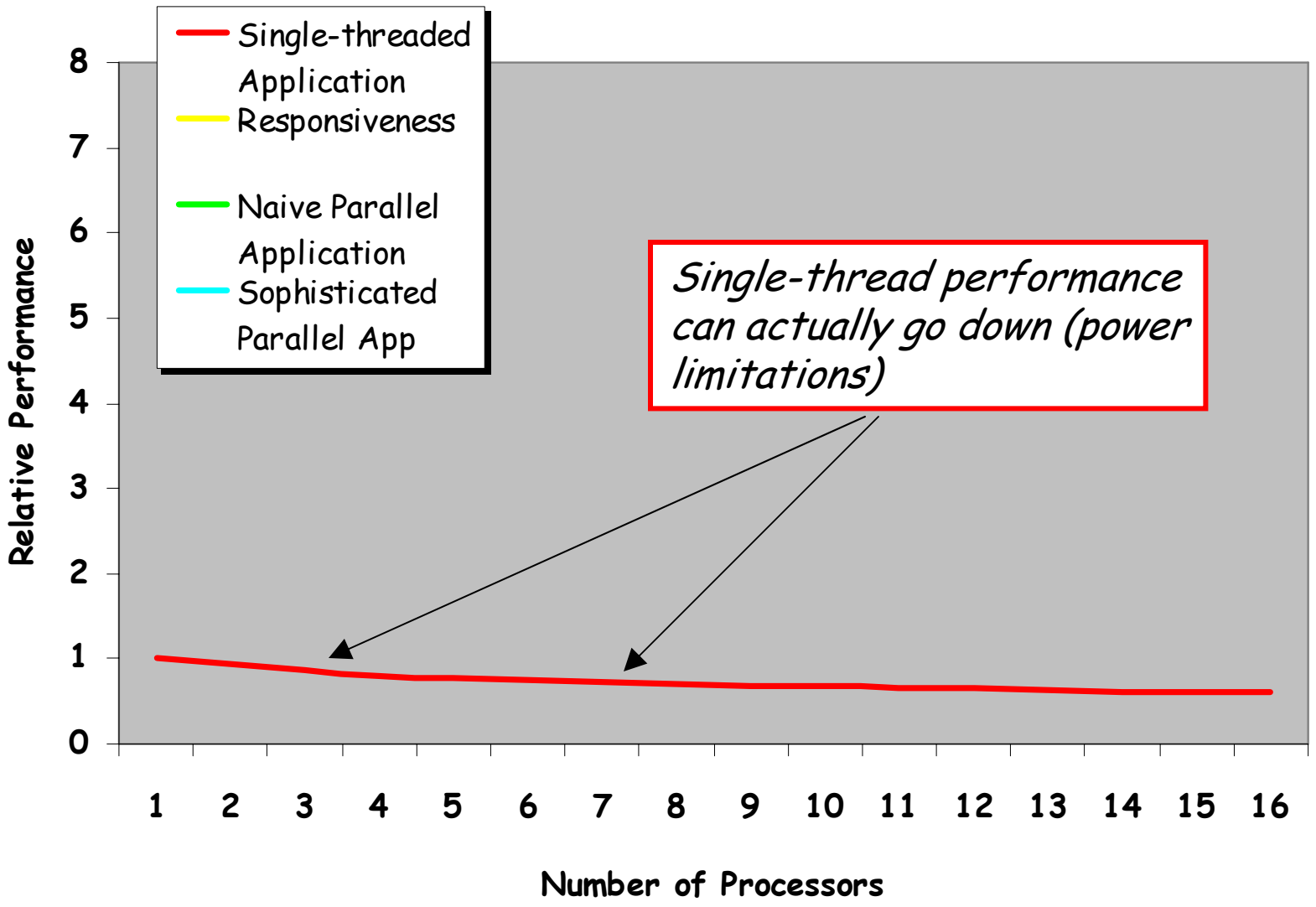
CMP Performance

(Hypothetical values)



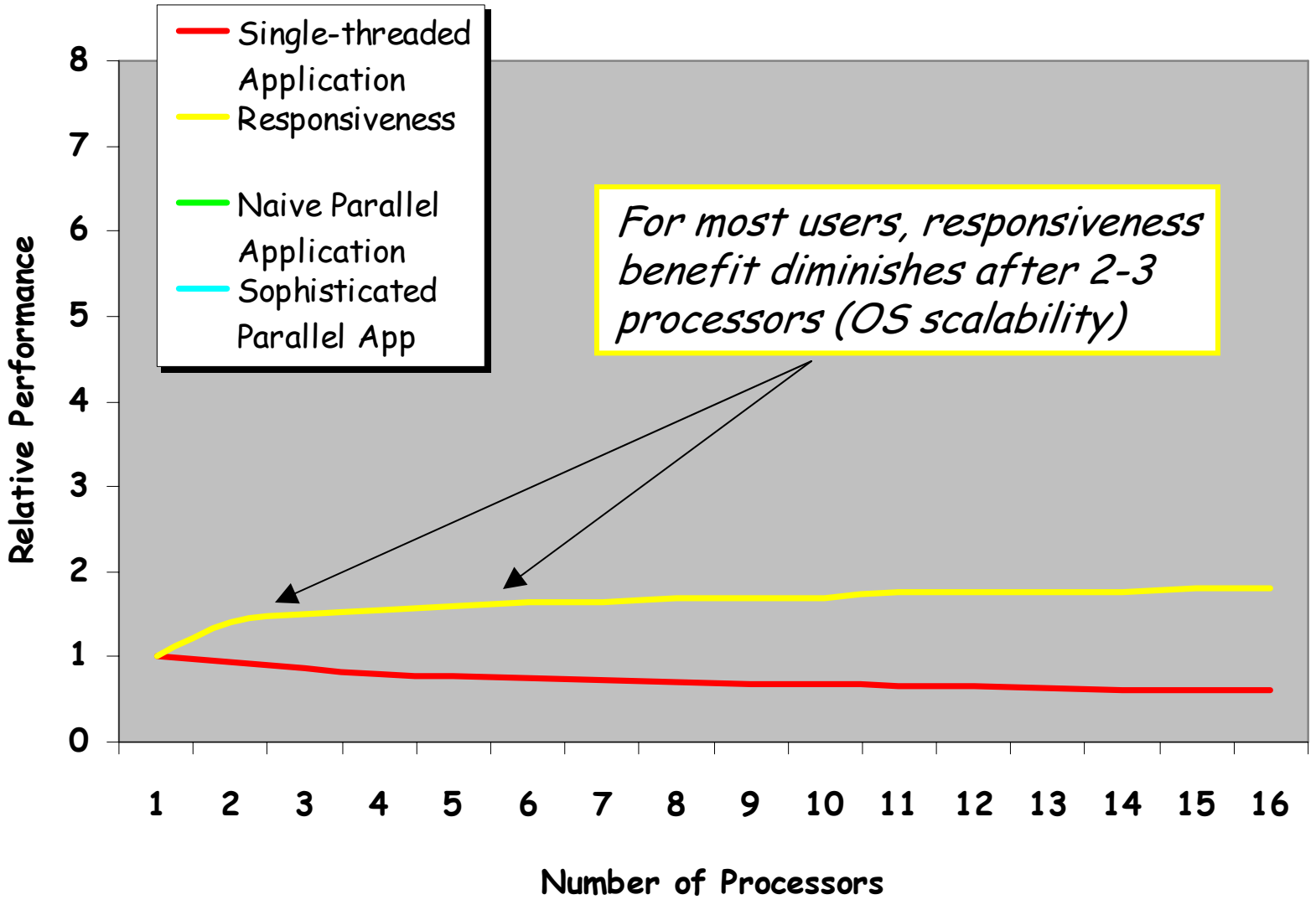
CMP Performance

(Hypothetical values)



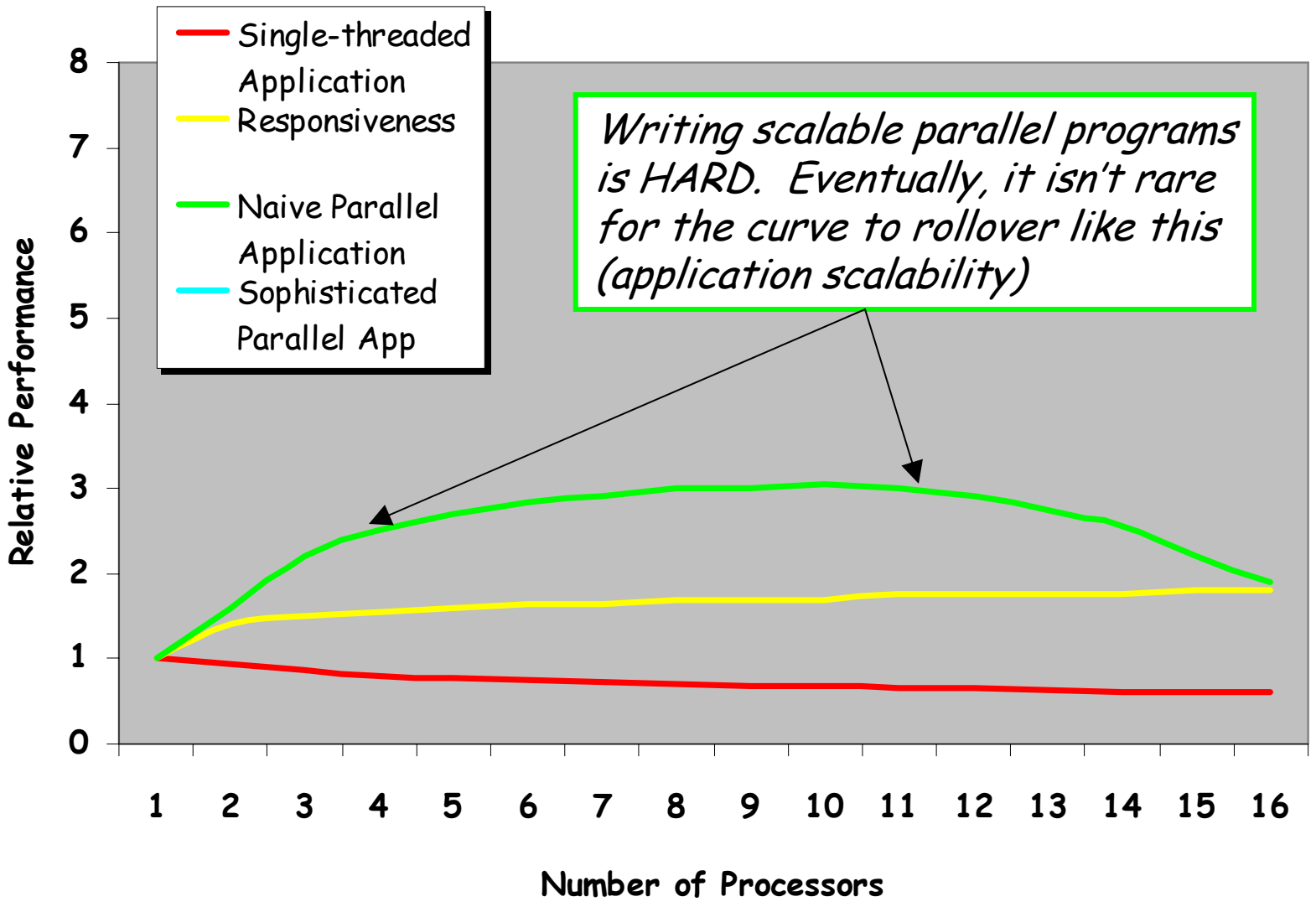
CMP Performance

(Hypothetical values)



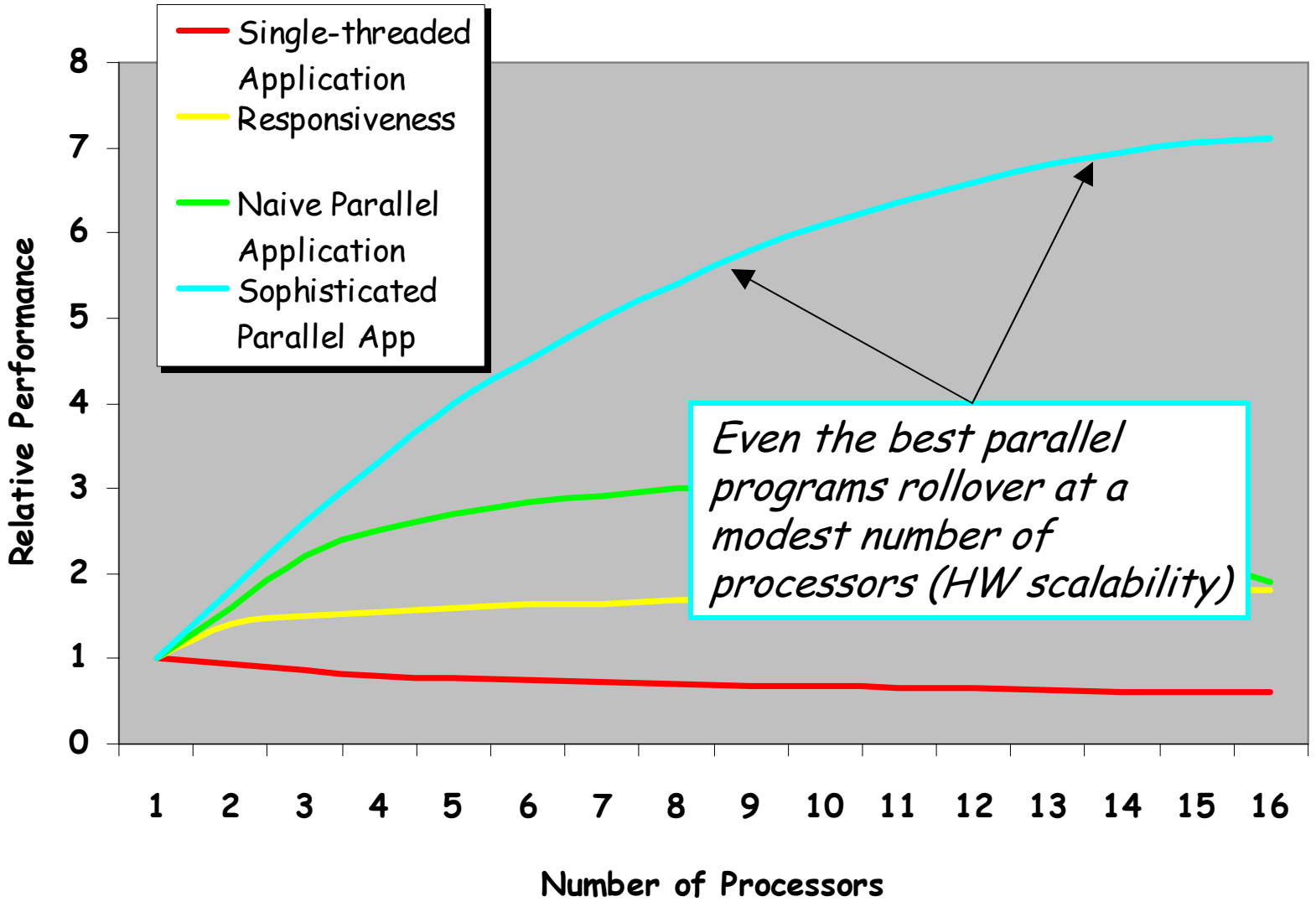
CMP Performance

(Hypothetical values)



CMP Performance

(Hypothetical values)



A Closer Look at *Parallelism*

Instruction-level Parallelism (ILP)

- Executing multiple instructions from same program at the same time
- Superscalar hardware picks up most available ILP

Data-level Parallelism (DLP)

- Executing same instruction on multiple pieces of data at the same time
- SSE hardware operates in this manner
- Also the basis for Vector Processors

Thread-level Parallelism (TLP) – several types:

1. **Concurrent Applications**
   - Multiple programs running at the same time
   - Multiple OS's on virtualized hardware image
2. **Internet Transactional**
   - Multiple computers running the same application
3. **Parallel Applications**
   - Single application partitioned to run as multiple threads
   - ***The holy grail of computer science***

The Transition to Parallel Applications

Single-threaded Applications

- Most of today's applications
- Well understood optimization techniques
- Advanced development, analysis and debug tools
- Conceptually, easy to think about

Parallel Applications

- Small number of applications (worked by experts for 10+ yrs)
- Awkward development, analysis and debug environments
- Parallel programming is hard!
- Amdahl's law is still a law
- SW productivity is already in a crisis → ***this worsens things!***

Understanding the *appropriate rate of transition* is what will ultimately be important

Hardware Considerations & Issues

- Think from the ***system-level inward***
  - Good designs are not just multi-core...
  - On-chip *system architecture*
- Balance is key
  - Old bottlenecks can get *n-times worse*
- Interference
  - **Constructive** – *cache sharing*
  - **Destructive** – *cache contention, bus contention, queuing delays, (M/T: internal core resources)*

First Three Generations of CMP

- ***Gen1***
  - Hasty integration of a bunch of cores
  - *Who would ever do such a thing?*
- ***Gen2***
  - Balanced integration of cores and system functionality
  - Address ***HW scalability*** inhibitors
- ***Gen3***
  - Chip-level framework & infrastructure to support wide range of CMP variants
  - Features to improve ***SW visibility*** & ***optimization*** for parallel application development

Generation 4 and beyond

- ***Gen4***
  - Runtime adaptive flexibility
  - Specialized resources and optimizations
  - Engage resources appropriate for each application
  - Introspective self-management

- ***Beyond Gen4?***
  - That will cost you a couple of beers ...